2003 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

THE VIRTUAL TEST BED PROJECT

Luis Rabelo, Ph.D.
Associate Professor
Industrial Engineering and Management Systems Department
University of Central Florida

KSC Colleague: Cary Peaden
KSC-YA-E6
KSC-YA

ABSTRACT

This is a report of my activities as a NASA Fellow during the summer of 2003 at the NASA Kennedy Space Center (KSC). The core of these activities is the assigned project: the Virtual Test Bed (VTB) from the Spaceport Engineering and Technology Directorate. The VTB Project has its foundations in the NASA Ames Research Center (ARC) Intelligent Launch & Range Operations program (ILRO). The objective of the VTB project is to develop a unique collaborative computing environment where simulation models can be hosted and integrated in a seamless fashion. This collaborative computing environment will have as emphasis operational models. This report will focus on the decisions about the different simulation modeling environments considered, simulation platform development, technology and operational models assessment, and computing infrastructure implementation.

THE VIRTUAL TEST BED PROJECT

Luis C. Rabelo

1. INTRODUCTION

The contributions to the Virtual Test Bed (VTB) Project during this fellowship are extensive. These contributions can be classified into the following areas: (1) systems architecting activities, (2) evaluations of models and processes, (3) evaluations of technologies, (4) systems design, (5) documentation and technology transfer, and (6) liaison with Academic and Industry groups.

This report is organized in the following sections. Section 2 introduces general concepts about complex systems, spaceports, and virtual test beds. These concepts will be needed to understand the objectives of the project and its implications for the future of NASA. The "preliminary architecture" is the topic of Section 3. The preliminary architecture of the VTB defines the different systems required in order to achieve the objectives of the Intelligent Launch & Range Operations (ILRO) project. Section 4 discusses the selection of the discrete-event simulation platform and the respective model to be hosted. The model to be hosted is the NASA Shuttle Simulation Model built by Mollaghasemi (Industrial Engineering and Management Systems Department – University of Central Florida), and Cates and Steele (NASA KSC). Section 5 discusses the models and process evaluations activities accomplished during this fellowship. Several models are being evaluated to see the opportunities to be integrated in the VTB. Section 6 discusses briefly some of the possible visualization schemes for the VTB.

2. GENERAL CONCEPTS

Spaceports are complex systems. According to Barth [1] "Spaceport technologies must employ a life-cycle "system of systems" concept in which major spaceport systems – launch vehicle processing systems, payload processing systems, landing and recovery systems, and range systems – are designed concurrently with flight vehicle systems and flight crew systems." Therefore, it seems logically to think that a virtual test bed can host the different models that represent different systems and elements of a spaceport. These models in the virtual test bed will work in an integrated fashion synthesizing in a holistic view and becoming together a Virtual Spaceport. This Virtual Spaceport can be utilized to test new technologies, new operational processes, the impact of new space vehicles in the spaceport supply chain, and the introduction of higher schemes of decision-making. A Virtual Spaceport will allow an intelligent visualization of the entire spaceport concept and the implementation of knowledge management strategies.

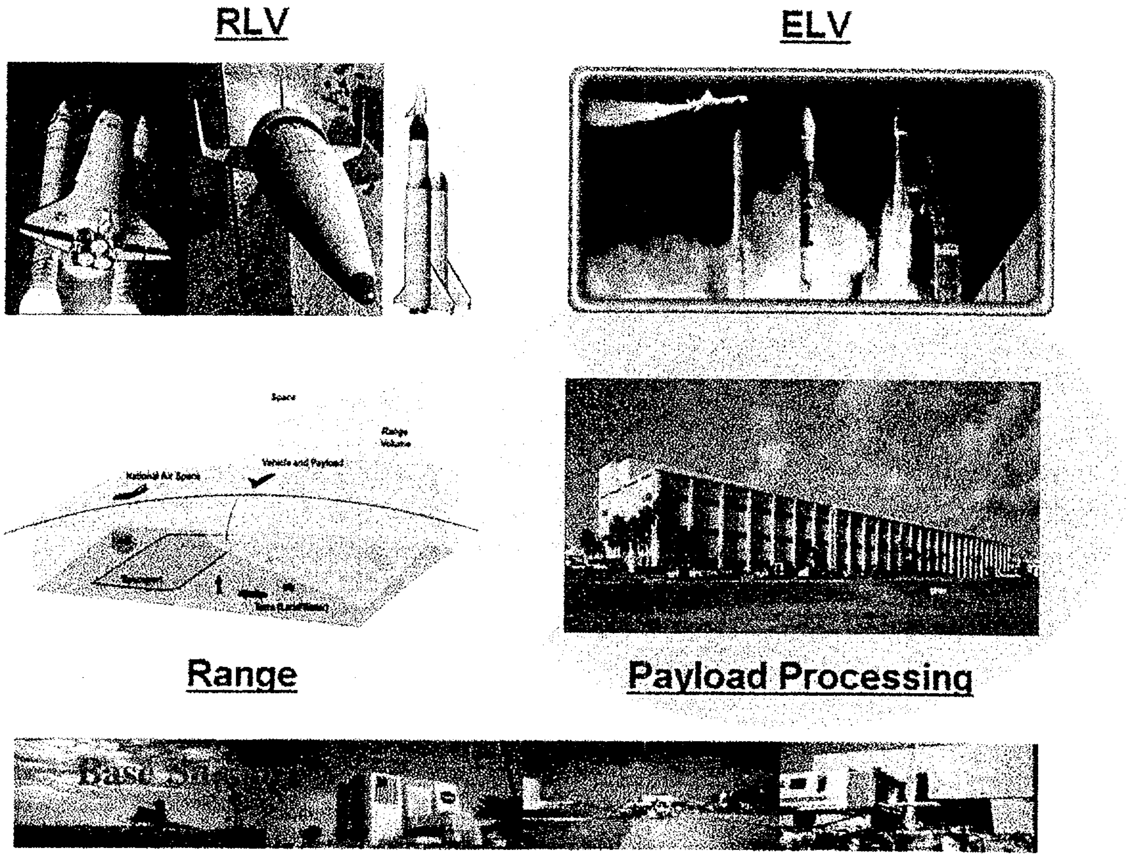

Figure 1. A spaceport is a system of systems

The Intelligent Launch and Range Operations (ILRO) Project at NASA Ames Research Center (ARC) was started to perform initial studies of a test bed with a demonstration. An evolution of the ILRO test bed is the Virtual Test Bed (VTB) Project. The objective of the Virtual Test Bed (VTB) Project is to provide a collaborative computing environment to support simulation scenarios, reuse, and integration of multidisciplinary models that represent elements of operations at spaceports. This VTB will be used

3. PRELIMINARY SYSTEMS ARCHITECTURE

One of the first activities was the development of an architecture. It is very well known that systems architecting integrates systems theory and systems engineering with architecting theory and practice of architecting [2,3]. Conceptualization is the keyword for architecting. System conceptualization involves creativity and the recognition of potential users and perceived needs. System architectures are driven by the function, instead of the form, of the system. Systems engineering is the one that provides the form.

The VTB is composed of the Integration User Interface, Decision-Maker User Interface, Security Component, Integration System, Simulation System, Model Functions Manager, Model Library Manager, Database System, and File Storage System. The Integration User Interface provides the capability to transfer a model to the VTB. The user can integrate an existing model (and create extensions to it) using the different tools and methodologies provided by this interface. The Decision-Maker User Interface is the simulation interface. The Decision-Maker User Interface supports the execution and the development of scenarios with the existing integrated models hosted on the VTB. The Security Component provides different levels of computer security such as password schemes, authentication, firewalls, Secure Socket Layer (SSL) implementations, maintenance and prevention mechanisms (e.g., anti-virus), certificates, and encryption. The Integration System takes the representation and information outlined by the engineer (using the Integration User Interface) and formulates a hierarchical description of entities, activities, and interactions that will be represented in an integrated model. The Simulation System executes the integrated model(s) according to the scenarios submitted by the decision-maker. The Simulation System invokes the integrated model(s) from the VTB Host and the model's operation functions from the Model Functions Manager. The Model Functions Manager provides the business logic for the different transactions to save the different model configurations as specified by the Integration System. The Model Functions Manager also retrieves from the Database System and the File Storage System the simulation models, data, and configuration parameters needed by the Simulation System. The Model Library Manager will support the development and management (retrieval, saving, configuration management) of the libraries. The Database System will store the model and its details. Finally, the File Storage System stores the model and its details in a scheme appropriate for facilitating the operations of the Simulation System and the interface with NASA Ames Research Center Intelligent Launch and Range Operations (ILRO) Virtual Test Bed.

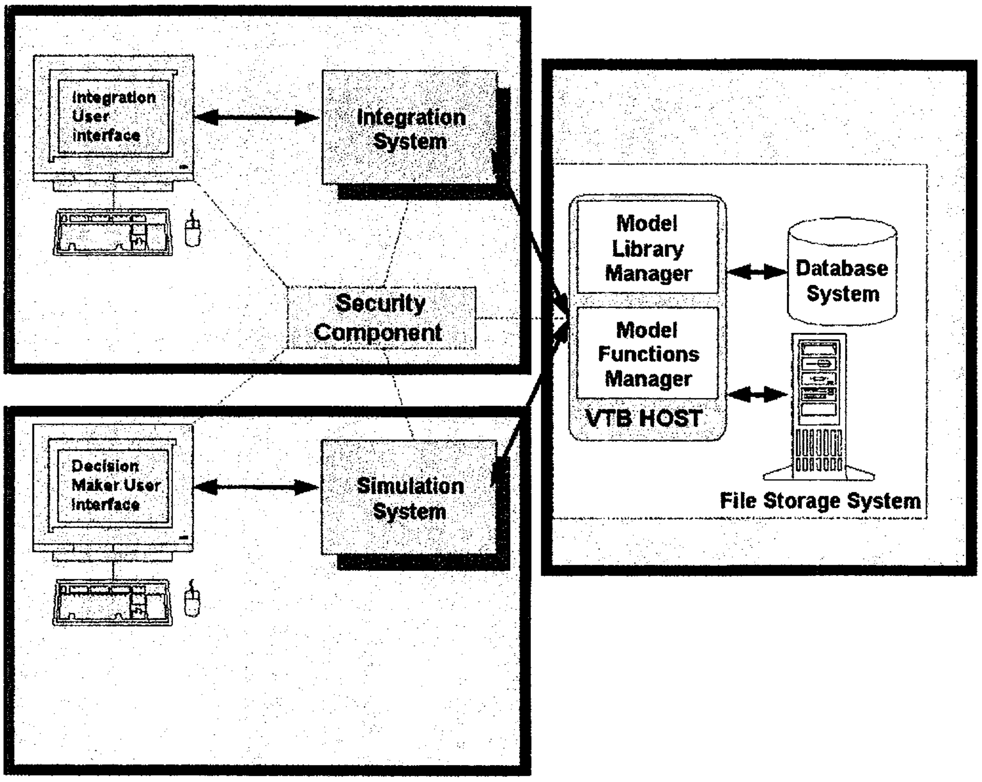

Figure 2. Preliminary systems architecture [4]

The VTB-SM will provide a computing environment capable of hosting in an integrated fashion different models that represent spaceport elements. This integration will allow managers, their staffs, and the aerospace community to examine concepts of operation, techniques, and procedures as an integral part of a spaceport through the use of human-in-control experiments. The VTB is intended to provide a robust, flexible, easy-to-use architecture, which can incorporate current and evolving operational characteristics and scenarios to conduct investigations. Where components-off-the shelf (COTS) software products can meet task requirements safely, the COTS software is utilized instead of developing custom applications. The software to be developed will be written in high-level languages such as Java, C, and C++, which have demonstrated a high degree of portability between platforms. This strategy provides a reliable system that is modular, expandable, and extensible. It is based on open hardware and software standards, easily incorporates new technology and user developed applications, and provides inherent user interface improvements.

The Simulation System will provide an environment to execute integrated simulators/models developed for specific elements of space operations into interactive simulator networks to support a single view of operations. For instance, NASA KSC has existing models that have been developed over time by different sources. These existing models have been developed from different points of view and for different aspects of the operation cycle. Consequently, existing models represent different levels of resolution and have selected different representation methods for internal entities, activities and interactions.

The Simulation System will employ object models and object-oriented methods to exercise a hierarchical description of entities, activities, and interactions represented in the integrated models. Figure 5 depicts a conceptualization of the functionality of the Simulation System using the High Level Architecture (HLA). The Simulation System shall follow the standards of the Department of Defense (DOD) and the Institute of Electrical and Electronic Engineers (IEEE) for the integration of models. The High Level Architecture (HLA) is one of those standards. HLA will be utilized to provide a consistent approach and rules for integrating distributed, heterogeneous, and legacy simulation systems. The HLA has been approved as an IEEE standard (http://standards.ieee.org/) and it has adopted as the facility for distributed simulation systems by the Object Management Group (http://simsig.omg.org/). The Run Time Infrastructure (RTI) software, which implements the rules and specifications of the HLA, provides methods, which can be called and used by individual simulation federates. RTI's interfaces can integrate federates, but implementation is quite complex.

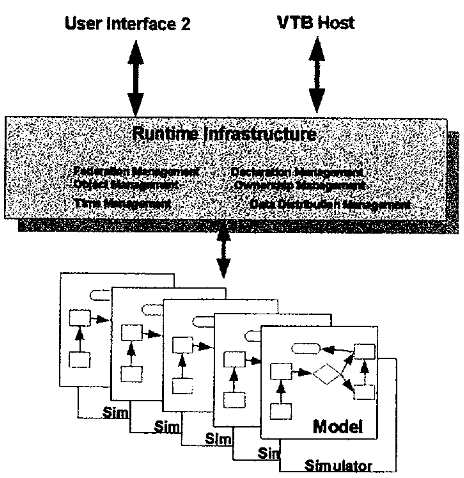

Figure 3. Conceptualization of the simulation system.

4. DISCRETE-EVENT SIMULATION ENVIRONMENT AND MODEL

A spaceport can only be represented using different types of models (sizes and nature). The natures of the models that the Simulation System will execute in an integrated fashion are: (1) Discrete-event models and (2) System Dynamics models. The simulation system will be a subsystem that will evolve over time to meet this important requirement. The scope of the first version to be delivered at the end of FY03 will focus on the hosting of a single discrete-event simulation model [4].

It was decided that the first version of the Simulation System will focus on discrete-event simulation. Therefore, we needed to find a discrete-event simulation environment capable (1) to handle multiple models, (2) compatible with HLA, (3) Open Source (to allow for modifications), (4) cost effective for NASA (to avoid the paying of expensive licenses and maintenance fees), and (5) a proven system being utilized as the backbone of advanced simulation environment. This environment will be demonstrated by the hosting a discrete-event model that represents a comprehensive representation of the NASA Shuttle operations. The NASA Shuttle Simulation Model is being hosted in the first version of the VTB to be delivered by the end of FY03. The NASA Shuttle Simulation Model is a simulation model for the operational life cycle of the Space Shuttle flight hardware elements through their respective ground facilities at KSC developed by NASA (Cates, Steele) and University of Central Florida (Mollaghasemi, Rabadi). The COTS tool used was Arena from Rockwell Software.

We decided to use as the backbone in discrete-event simulation for the HLA implementation the Synchronous Parallel Environment for Emulation and Discrete-Event Simulation (SPEEDES). SPEEDES allocates events over multiple processors to get simulation speed-up (www.speedes.com). This feature enhances runtime, especially when exploiting the very large number of processors and the high-speed internal communications found in high performance computing platforms. SPEEDES is HLA compliant. Another important characteristic of SPEEDES is object-orientation. SPEEDES' object-oriented architecture has a significant impact on the development of simulations. Entities in a system can be represented by individual classes. Such a representation, in turn, facilitates the distribution of the simulation models on different processors and the design of parallel simulation experiments. In addition, SPEEDES supports distributed simulation over the World Wide Web. This provides a very important advantage – a key feature of the World Wide Web for running a distributed simulation is the transparency of network heterogeneity, where interoperability of different networks is achieved through well-defined, standardized protocols such as HTTP and CGI.

A very important challenge for the future is the development of a hybrid simulation environment. Several studies of complex non-linear systems have shown the presence of non-stationary or even chaotic behavior in different operational regions of space vehicles. Discrete-event simulation models allow us to capture the system performance for a specific value of decision variables. However, they do not allow us to capture the dynamics of the system in the region or neighborhood of the control policy or decision variable values being evaluated. Hence, it would be desirable to develop modeling tools that work in conjunction with discrete-event simulation to allow us to evaluate the stability of a complex system in different operational regions. Simulation modeling of hybrid models will be needed to capture discrete and continuous states to accomplish higher levels of simulation fidelity. It is very clear that switching of modes occurs during the reentry transition phase of the NASA Shuttle. This hybrid simulation modeling will have to be studied.

5. ASSESSMENT OF MODELS

During the fellowship, we guided the evaluation of different operational models developed or being developed at KSC. Example of these models and modeling projects are:

- The NASA Shuttle Simulation Model (Current Shuttle Operations/Discrete-Event/COTS)
- GEM-FLO (Generic Environment for Simulation Modeling of Future Launch Operations/Discrete-Event/COTS)
- RPST (Range Process Simulation Tool/Decision Making/Customized Environment)
- ATLAS V (ELV operations by Lockheed/Discrete-Event/COTS)
- AATE (Cost Environment)
- Task Analysis (Project – being defined)
- VAB/OPF (3D Models and future discrete-event simulation environment using QUEST (DELMIA))
- MPLM (Models related to MPLM)
- VisionSpaceport Models (3d Models and Decision-Making Tool)

6.VISUALIZATION

The VTB will need good visualization capabilities. Several technologies such as VRML, X3D, and JAVA 3D were studied too (OpenGL was studied in Summer 2002).

- **Virtual Reality Modeling Language(VRML)**

VRML is a 3D content development language for the Internet. VRML is a high-level language that is used to describe 3D objects and scenes. VRML supports multiple platforms over the Internet.

- **Extensible 3D(X3D)**

X3D is an evolutionary step of VRML. X3D is designed to allow the best capabilities of VRML to be expressed in terms of XML.

- **Java 3D**

Java 3D is a 3D extension to Java. Java 3D allows developers to create comprehensive, platform-independent 3D applications and Web-based applets.

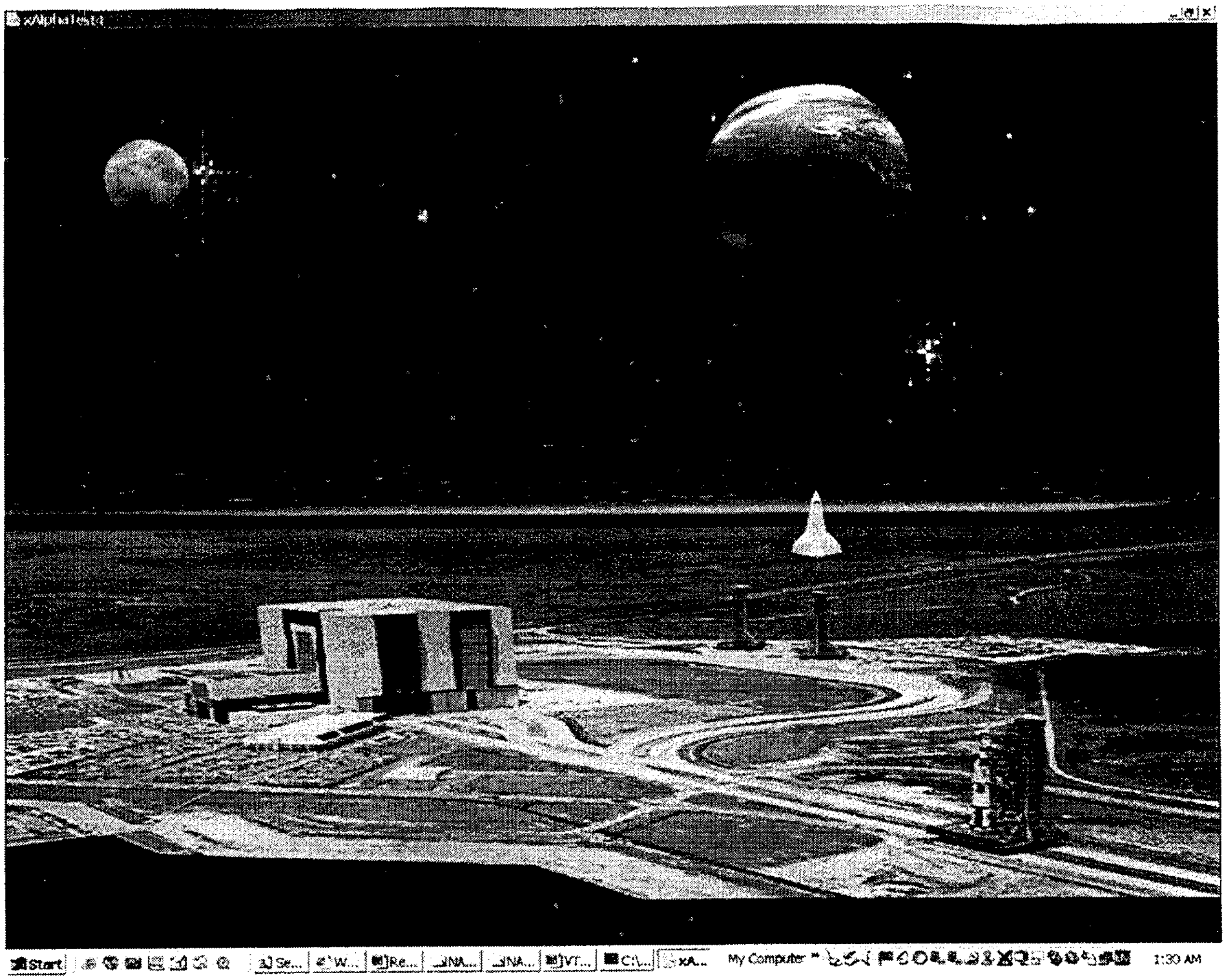

Figure 4. Example of application using JAVA 3D

REFERENCES

1. Barth, T., “Found in Space,” IEE Solutions, April 2002.
2. Rechtin, E., Systems Architecting, Prentice Hall, 1991.
3. Rechtin, E. and Maier, M., The Art of Systems Architecting, CRC Press, 1997.
4. VTB SLS Document, January 2003.